IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number:	10/733,037
Filing Date:	12/11/2003
Applicant(s):	Keith J. Purcell
Entitled:	METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR AUTOMATIC CODE GENERATION IN AN OBJECT ORIENTED ENVIRONMENT
Examiner:	Qing Chen
Group Art Unit:	2191
Attorney Docket No.:	RSW920030159US1 (7161-515U)

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is Appellant's Appeal Brief in support of the Notice of Appeal dated April 2, 2010. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, to Deposit Account 09-0461, and please credit any excess fees to such deposit account.

Date: July 19, 2010

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg
Registration No. 44,725
Customer Number 46320

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number:	10/733,037
Filing Date:	12/11/2003
Applicant(s):	Keith J. Purcell
Entitled:	METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR AUTOMATIC CODE GENERATION IN AN OBJECT ORIENTED ENVIRONMENT
Examiner:	Qing Chen
Group Art Unit:	2191
Attorney Docket No.:	RSW920030159US1 (7161-515U)

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed April 19, 2010, wherein Appellants appeal from the Examiner's rejection of claims 1, 4 through 9, 12 through 19, and 21 through 26.

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by assignment recorded on December 11, 2003 at Reel 014778, Frame 0389.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1, 4 through 9, 12 through 19, and 21 through 26 are pending in this Application, claims 10, 11 and 27 having been canceled in the amendment dated October 9, 2007 (the "Third Amendment") and claims 2, 3 and 20 having been canceled in the amendment dated December 3, 2008 (the "Sixth Amendment"). It is from the multiple rejections of claims 1, 4 through 9, 12 through 19, and 21 through 26 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Claims 1, 4 through 12, 16 through 19 and 21 through 26 were amended in the amendment dated November 21, 2006 (the "First Amendment"). Claims 1, 6 through 9, 12, 14 through 19 and 25 through 26 were amended in the amendment dated April 20, 2007 (the "Second Amendment"). Claims 1, 12, 15 through 19 and

21 through 26 were amended in the Third Amendment. Claims 1, 12 and 17 through 19 were amended in the amendment dated June 9, 2008 (the "Fourth Amendment"). Claims 1, 12 and 17 through 19 were amended again in the amendment dated October 14, 2008 (the "Fifth Amendment"). Claims 1, 12 and 19 were amended yet again in the Sixth Amendment. Claims 1, 8, 9 ,12, 15, 16, 17 , 18 and 19 were amended even yet again in the amendment dated June 16, 2009 (the "Seventh Amendment"). Finally, claims 1, 6 through 8, 12 and 15 through 19 were amended in the amendment dated September 28, 2009 (the "Eighth Amendment").

V. SUMMARY OF CLAIMED SUBJECT MATTER

With respect to claim 1, a method for automatically generating computer program code has been claimed. (Par. [0013], lines 1-2) The method includes receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer. (Par. [0014], lines 2-4 and Par. [0015], line 1) The method also includes parsing the description in the web service to identify object parameters for the computing application (Par. [0015], lines 1-3) and locating a coding module corresponding to at least one of the object parameters within a node contained within the computational grid coupled to the web service over a computer

communications network. (Par. [0015], lines 3-6) In this regard the computational grid includes computers sharing computational resources and different coding modules. (Par. [0012], lines 1-3) The description can be supplied to the node contained within the computational grid (Par. [0016], lines 1-2) and applied to the located coding module to generate at least one output object corresponding to the identified object parameters. (Par. [0016], lines 2-3) Finally, the output object can be returned to the author over the computer communications network. (Par. [0017], lines 1-2)

With respect to claim 12, a computer program product is embodied on a computer readable medium for automatically generating computer program code. (Par. [0010], lines 1-4) The computer includes computer executable instructions for performing a method that includes receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer. (Par. [0014], lines 2-4 and Par. [0015], line 1) The method also includes parsing the description in the web service to identify object parameters for the computing application (Par. [0015], lines 1-3) and locating a coding module corresponding to at least one of the object parameters within a node contained within the computational grid coupled to the web service over a computer communications network. (Par. [0015], lines 3-6) In

this regard the computational grid includes computers sharing computational resources and different coding modules. (Par. [0012], lines 1-3) The description can be supplied to the node contained within the computational grid (Par. [0016], lines 1-2) and applied to the located coding module to generate at least one output object corresponding to the identified object parameters. (Par. [0016], lines 2-3) Finally, the output object can be returned to the author over the computer communications network. (Par. [0017], lines 1-2)

With respect to claim 19, system for automatically generating computer program code includes a computational grid of computers (Par. [0011], lines 1-4) sharing computational resources and including different nodes, (Par. [0012], lines 1-2) each node including at least one programming model. (Par. [0012], lines 2-4) The system further includes a web service coupled to the web service over a computer communications network for receiving an application description from an author from over the computer communications network, (Par. [0014], lines 2-4 and Par. [0015], line 1) for parsing the application description to identify object parameters for a computing application, (Par. [0015], lines 3-6) to locate a coding module corresponding to at least one of the object parameters within a node residing on the computational grid, (Par. [0015], lines 3-6) for supplying the application description to the node in which the located coding module applies the

application description (Par. [0016], lines 1-2)to generate at least one output object corresponding to the identified object parameters in the application description, (Par. [0016], lines 2-3) and returning the output object to the author over the computer communications network. (Par. [0017], lines 1-2)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The rejection of claims 1, 4 through 9, 12 through 19 and 21 through 26 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication No. 2004/0088688 by Hejlsberg et al. (Hejlsberg) in view of U.S. Patent No. 7,185,046 to Ferstl et al. (Ferstl)

VII. THE ARGUMENT

THE REJECTION OF CLAIMS 1, 4 - 9, 12 - 19 AND 21 - 26 UNDER 35 U.S.C. § 103

For the convenience of the Honorable Board, claims 4 through 9 stand or fall together with claim 1, claims 13 through 18 stand or fall together with claim 12 and claims 21 through 26 stand or fall together with claim 19.

Each of claims 1, 12 and 19 pertain to the automated generation of computer program code and refer to the providing of a description of a computing application to a Web service that in turn, selects a coding module in a node within

a computational grid corresponding to parameters identified in the description so that the relevant computer program code can be generated. Exemplary claim 1 provides:

> 1. A method for automatically generating computer program code comprising the steps of:
>
> receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer;
>
> parsing said description in said web service to identify object parameters for said computing application;
>
> locating a coding module corresponding to at least one of the object parameters within a node contained within the computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules;
>
> supplying said description to said node contained within the computational grid;
>
> applying said description to said located coding module to generate at least one output object corresponding to the identified object parameters; and
>
> returning said at least one output object to the author over the computer communications network.

Thus, integral to each independent claim is the location of a coding module in a node within the computational grid that corresponds to identified object parameters.

As noted on pages 10 and 11 of the Eighth Amendment, rationale (A) of the Examination Guidelines of M.P.E.P. 2141 admittedly employed by Examiner in

establishing a prima facie case of obviousness with respect to claims 1, 12 and 19, requires that Examiner locate within the combination of Ferstl and Hejlsberg all claimed limitations of claims 1, 12 and 19. As such, at issue is whether or not the combination of Ferstl and Hejlsberg teach every claimed limitation of claims 1, 12 and 19 including the integral limitation of locating a coding module in a node within the computational grid that corresponds to identified object parameters so as to render claims 1, 12 and 19 obvious under the law. On pages 6 and 7 of the non-final office action dated January 19, 2010 (the "Last Non-Final Office Action"), Examiner alleges to have found the integral limitation set forth above in Ferstl.

In particular, Examiner argues at page 7 of the Last Non-Final Office Action that column 1, line 52-59 and 65-67, column 2, lines 1-8 and column 12, lines 25-31 provide the relevant teaching. Examiner specifically states,

> However, Hejlsberg does not disclose:
>
> locating a coding module corresponding to at least one of the object parameters within a node contained within a computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules.
>
> Ferstl discloses:
>
> locating a coding module corresponding to at least one of object parameters within a node contained within a computational grid coupled to a web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules *(see Column 1: 52-59, "A computing grid is a hardware and software infrastructure serving to*

handle computing jobs submitted by a user. The computing grid may interconnect distributed computers, storage devices, mobile devices, instruments, sensors, data bases and/or software applications. Generally a computing grid may comprise virtually any kind of computing device and includes a grid infrastructure to handle the distribution of computing jobs." and 65-67 to Column 2: 1-8, *"Upon receiving an instruction to distribute a computing job the grid infrastructure selects a suitable computing device and transfers the computing job to the selected computing device." and "Accordingly, a user or application at a client device may issue an instruction to execute a computing job towards the grid infrastructure which in turn selects a suitable processing element and the processing results are ultimately returned to the client. "; Column* 12: 25-31, *"In an example, the selection section obtains the selection information and accesses data specifying corresponding features of a plurality of job handlers, for example, in a configuration file specifying features of a plurality of job handlers available. Then, the selection section may identify a suitable job handler matching the selection information in association with the job request. ").*

Thus, Examiner's analysis appears to be an aggregation of the verbatim claim language of Appellants' claim 1 and the verbatim text of the cited portions of Ferstl without any significant analysis of how Examiner has construed critical claim limitations such as "coding module" or "locate" or "object parameters", and how Examiner has applied the teachings of Ferstl to the construed claim language.

To Examiner's credit, however, on page 17 of the Last Non-Final Office Action, after repeating again verbatim Appellants' claim language of the claim limitation in dispute and the text of Ferstl. In this regard, Examiner states at page 17 that a computing grid is both hardware and software interconnecting, among other things, software applications and that "the computing grid selects a suitable processing element to process a computing job". Regrettably, as for the ONLY substantive analysis provided by Examiner, Examiner provided NO discussion of a

teaching in Ferstl that directly maps to the location of a coding module corresponding to one or more object parameters within a node contained within a computational grid coupled to a web service as claimed by Appellants in each of claims 1, 12 and 19.

As noted by the Supreme Court in Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.,[9] a clear and complete prosecution file record is important in that "[p]rosecution history estoppel requires that the claims of a patent be interpreted in light of the proceedings in the USPTO during the application process." The Courts that are in a position to review the rejections set forth by the Examiner (i.e., the Board of Patent Appeals and Interferences, the Federal Circuit, and the Supreme Court) can only review what has been written in the record; and therefore, the Examiner must clearly set forth the rationale for the rejection and clearly and particularly point out those elements within the applied prior art being relied upon by the Examiner in the statement of the rejection.

This requirement that the Examiner clearly set forth the rationale for the rejection and clearly and particularly point out those elements within the applied prior art being relied upon by the Examiner in the statement of the rejection is found in with 37 C.F.R. § 1.104(c), which reads:

> In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command. When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable. The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.

Moreover, in the unpublished opinion of Ex parte Pryor[10], the Board of Patent Appeals and Interferences recognized the necessity for an Examiner to supply sufficient information to establish a prima facie case of anticipation. Specifically, the Board wrote:

> At the outset, we note the examiner has been of little help in particularly explaining the rejections on appeal. A mere statement that claims stand rejected "as being clearly anticipated by" a particular reference, without any further rationale, such as pointing out corresponding elements between the instant claims and the applied reference, fails to clearly make out a prima facie case of anticipation. (emphasis in original)

Despite the very specific requirement for the Examiner to clearly set forth the rationale for the rejection and clearly and particularly point out those elements within the applied prior art being relied upon by the Examiner, the Examiner has failed to do so with respect to setting forth a clear rationale. Instead, the Examiner's statement of the rejection simply consists of the Examiner repeating, almost word-for-word, each of the recited claims and asserting that the entire claim is disclosed by certain specified passages within Ferstl.

The importance of the specificity requirement of 37 C.F.R. § 1.104(c) is also further evident in M.P.E.P. § 706.07, which states:

> The examiner should never lose sight of the fact that in every case the applicant is entitled to a full and fair hearing, and that a clear issue between applicant and examiner should be developed, if possible, before appeal.

A clear issue, however, cannot be developed between Applicants and the Examiner where the basis for the Examiner's rejection of the claims is ambiguous. The Examiner's "analysis" provides little insight as to (i) how the Examiner is interpreting the elements of the claims and (ii) what specific features within Ferstl the Examiner believes identically discloses the specific elements (and interactions between elements) recited in the claims. By failing to specifically identify those features within Ferstl being relied upon in the rejection, the Examiner has essentially forced Appellants to engage in mind reading and/or guessing to determine how the Examiner is interpreting the elements of the claims and what specific features within Ferstl the Examiner believes identically disclose the claimed invention.

Nevertheless, a thorough review of the cited portions of Ferstl reveal only the teaching of a computational grid and the selection of different nodes to process different jobs of a distributed computing operation. (See column 2, lines 1 through 8) The teachings reproduced by Examiner on pages 6 and 7 of the Last Non-Final Office Action do not attempt to account for the location of a CODING MODULE that corresponds to OBJECT PARAMETERS within a node of a computational grid.

Thus, under rationale (A) of the Examination Guidelines, Examiner has failed to account for all claimed limitations of claims 1, 12 and 19 and thus has not established a prima facie case of obviousness under 35 U.S.C. § 103(a).

Date: July 19, 2010

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg
Registration No. 44,725
Customer Number 46320

VIII. CLAIMS APPENDIX

1. (Previously Amended) A method for automatically generating computer program code comprising the steps of:

receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer;

parsing said description in ~~by~~ said web service to identify object parameters for said computing application;

locating a coding module corresponding to at least one of the object parameters within a node contained within the computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules;

supplying said description to said node contained within the computational grid;

applying said description to said located coding module to generate at least one output object corresponding to the identified object parameters; and

returning said at least one output object to the author over the computer communications network.

2. (Cancelled)

3. (Cancelled)

4. (Previously Presented) The method as set forth in claim 1, wherein said description is generated using Object Meta Language (OML).

5. (Previously Presented) The method as set forth in claim 4, wherein said OML is an eXtensible Markup Language (XML) dialect.

6. (Previously Amended) The method as set forth in claim 1, wherein said located coding module is an XML template.

7. (Previously Amended) The method as set forth in claim 1, wherein said located coding module is an eXtensible Stylesheet Language (XSL) style sheet.

8. (Previously Amended) The method as set forth in claim 7, wherein the step of applying said description to said located coding module further comprises the steps of:

parsing said description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable,

wherein said at least one replacement variable is the result of an XML/XSL transform.

9. (Previously Amended) The method as set forth in claim 6, wherein the step of applying said description to said located coding module further comprises the steps of:

parsing said description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is stored in said XML template.

10. (Cancelled)

11. (Cancelled)

12. (Previously Amended) A computer program product embodied on a computer readable medium for automatically generating computer program code, comprising computer executable instructions for:

receiving from an author over a computer communications network a description of a computing application in a web service executing in memory by a processor in a computer;

parsing said description in ~~by~~ said web service to identify object parameters for said computing application;

locating a coding module corresponding to at least one of the object parameters within a node contained within the computational grid coupled to the web service over a computer communications network, the computational grid comprising a plurality of computers sharing computational resources, said computational grid further comprising a plurality of coding modules;

supplying said description to said node contained within the computational grid;

applying said description to said located coding module to generate at least one output object corresponding to the identified object parameters; and

returning said at least one output object to the author over the computer communications network.

13. (Original) The computer program product as set forth in claim 12, wherein said description comprises Object Meta Language (OML).

14. (Previously Presented) The computer program product as set forth in claim 13, wherein said OML is an eXtensible,Markup Language (XML) dialect.

15. (Previously Amended) The computer program product as set forth in claim 12, wherein said coding module is an XML template.

16. (Previously Amended) The computer program product as set forth in claim 12, wherein said coding module is an eXtensible Stylesheet Language (XSL) style sheet.

17. (Previously Amended) The computer program product as set forth in claim 15, wherein the computer executable instructions for applying said description to said located coding module further comprise instructions for:

parsing said description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is the result of an XML/XSL transform.

18. (Previously Amended) The computer program product as set forth in claim 15, wherein the computer executable instructions, for applying said description to said located coding module further comprise instructions for:

parsing said description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable, wherein said at least one replacement variable is stored in said XML template.

19. (Previously Amended) A system for automatically generating computer program code comprising:

a computational grid, wherein said computational grid includes a plurality of computers sharing computational resources, said grid comprising a plurality of nodes, each node comprising at least one programming model; and

a web service coupled to the web service over a computer communications network for receiving an application description from an author from over the computer communications network, for parsing said application description to identify object parameters for a computing application, to locate a coding module corresponding to at least one of the object parameters within a node residing on said computational grid, for supplying said application description to said node in which said located coding module applies said application description to generates at least one output object corresponding to the identified object parameters in said

application description, and returning said at least one output object to the author over the computer communications network.

20. (Cancelled)

21. (Previously Presented) The system as set forth in claim 19, wherein said application description is generated using Object Meta Language (OML).

22. (Previously Presented) The system as set forth in claim 21, wherein said OML is an eXtensible Markup Language (XML) dialect.

23. (Previously Amended) The system as set forth in claim 19, wherein said coding modules are XML templates.

24. (Previously Amended) The system as set forth in claim 19, wherein said coding modules are eXtensible Stylesheet Language (XSL) style sheets.

25. (Previously Amended) The system as set forth in claim 24, wherein said coding modules for generating an application description includes computer code for:

parsing said application description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable,

wherein said at least one replacement variable is the result of an XML/XSL transform.

26. (Previously Amended) The system as set forth in claim 23, wherein said coding modules for generating an object from said application description include computer code for:

parsing said application description to locate at least one variable; and

substituting said at least one variable with at least one replacement variable,

wherein said at least one replacement variable is stored in said XML template.

27. (Cancelled)

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.